AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A method of encoding three-dimensional object data, which is comprised of point texture data, voxel data, or octree structure data, the method comprising:

generating three-dimensional object data using a three-dimensional bounding volume to convert the three-dimensional object data into voxel data, wherein the voxels are differentiated based on whether they are located where objects exist or in a background;

representing the voxel data by a tree structure of a predetermined depth in which nodes include attached labels indicating their respective types, the types comprising nodes having sub-nodes, nodes having all voxels located in the background, nodes having all voxels located where objects exist, and nodes at the predetermined depth having voxels located where objects exist and in the background, wherein the nodes at the predetermined depth having voxels located where objects exist and in the background do not have sub-nodes;

encoding nodes of the three-dimensional object data; and

generating the three-dimensional object data whose nodes are encoded into a bitstream.

2. (Canceled)

3. (Previously Presented) The method of claim 1, wherein in the tree structure representing the three-dimensional object data, a node having sub-nodes is labeled 'S', a node whose voxels are all located in the background is labeled 'W', a node whose voxels are all located where objects exist is labeled 'B', and a node at the predetermined depth whose voxels are located where objects exist and in the background is labeled 'P', wherein 'P' nodes are encoded using a prediction-by-partial-matching (PPM) algorithm.

4. (Previously Presented) The method of claim 3, wherein encoding the nodes of the three-dimensional object data comprises:

encoding node information which indicates whether or not a current node is an 'S' node or a 'P' node; and

encoding detailed information bit (DIB) data of an 'S' node if the node information indicates that the current node is an 'S' node and encoding DIB data of a 'P' node if the node information indicates that the current node is a 'P' node.

5. (Original) The method of claim 1, wherein in encoding the nodes of the three-dimensional object data, only some of the nodes of the three-dimensional object data, ranging from a root node to a predetermined lower node, are encoded.

6. (Original) The method of claim 1, wherein in encoding the nodes of the three-dimensional object data, all the nodes of the three-dimensional object data are encoded.

7. (Currently Amended) A method of encoding three-dimensional object data, which is comprised of point texture data, voxel data, or octree structure data, the method comprising:

(a) generating three-dimensional object data having a tree structure of a predetermined depth in which nodes include attached labels indicating their respective types, the types comprising nodes having sub-nodes, nodes having all voxels located in a background, nodes having all voxels located where objects exist, and nodes at the predetermined depth having voxels located where objects exist and in the background, wherein the nodes at the predetermined depth having voxels located where objects exist and in the background do not have sub-nodes;

(b) merging the nodes of the three-dimensional object data by referring to their labels;

(c) encoding merged nodes;

(d) generating the three-dimensional object data whose merged nodes are encoded into a bitstream; and

(e) repeatedly carrying out steps (a) through (d) until an uppermost node of the tree structure representing the three-dimensional object data is encoded.

8. (Previously Presented) The method of claim 7, wherein in step (a), a node having sub-nodes is labeled 'S', a node whose voxels are all located in the background is labeled 'W', a node whose voxels are all located where objects exist is labeled 'B', and a node at the predetermined depth whose voxels are located where

objects exist and in the background is labeled 'P', wherein 'P' nodes are encoded using a prediction-by-partial-matching (PPM) algorithm.

9. (Previously Presented) The method of claim 8, wherein step (b) comprises:

selecting 'P' nodes and 'S' nodes whose sub-nodes are labeled 'W' and 'B' as candidate nodes to be merged;

selecting, from among the candidate nodes as an optimal node, a node which can minimize a ratio of a difference ΔD between a number of distorted bits before merging the candidate nodes and a number of distorted bits after merging the candidate nodes with respect to a difference ΔR between a number of bits before merging the candidate bits and a number of bits after merging the candidate bits;

labeling the selected node 'B'; and

updating all the candidate nodes except the node selected as an optimal node.

10. (Original) The method of claim 9, wherein D is calculated in the following equation using a Hamming distance between an original model V and its approximation $\hat{V}$ as distortion measurement:

$$D=\sum_{x=1}^{X}\sum_{y=1}^{Y}\sum_{z=1}^{Z}\left|V(x,y,z)-\hat{V}(x,y,z)\right|$$

where X×Y×Z represents the resolution of the original model.

11. (Previously Presented) The method of claim 9, wherein step (c) comprises:

encoding a continue flag which includes information indicating whether or not the candidate nodes exist in a queue;

encoding node position information which indicates a position of each of the candidate nodes in the queue;

encoding node type information which indicates whether or not a current node is an 'S' node or a 'P' node; and

encoding detailed information bit (DIB) data of an 'S' node if the node type information indicates that the current node is an 'S' node and encoding DIB data of a 'P' node if the node type information indicates that the current node is a 'P' node.

12. (Original) The method of claim 11, wherein encoding the DIB data of the 'S' node comprises:

encoding an average of color information; and

encoding labels of eight sub-nodes of the current node.

13. (Previously Presented) The method of claim 11, wherein encoding the DIB data of the 'P' node comprises:

encoding depth information; and

encoding color information.

14. (Original) The method of claim 13, wherein in encoding the depth information, all nodes below a predetermined node in the tree structure representing

the three-dimensional object data are PPM-encoded according to a raster scanning order by using a predetermined number of contexts.

15. (Original) The method of claim 13, wherein in encoding the color information, red (R), green (G), and blue (B) values of 'B' voxels of the current node are encoded by carrying out differential pulse code modulation (DPCM) and adaptive arithmetic coding (AAC).

16. (Previously Presented) The method of claim 7, wherein in step (c), only some of the merged nodes, ranging from a first merged node to a predetermined numbered merged node, are encoded.

17. (Original) The method of claim 7, wherein in step (c), all the merged nodes are encoded.

18. (Currently Amended) An apparatus for encoding three-dimensional object data, which is comprised of point texture data, voxel data, or octree structure data, the apparatus comprising:

a tree structure generator which generates three-dimensional object data having a tree structure of a predetermined depth in which nodes include attached labels indicating their respective types, the types comprising nodes having sub-nodes, nodes having all voxels located in a background, nodes having all voxels located where objects exist, and nodes at the predetermined depth having voxels located where objects exist and in the background, wherein the nodes at the

predetermined depth having voxels located where objects exist and in the background do not have sub-nodes;

a merging order selector which merges the nodes of the three-dimensional object data by referring to their labels;

a node encoder which encodes merged nodes; and

a bitstream generator which generates the three-dimensional object data whose merged nodes are encoded into a bitstream.

19. (Previously Presented) The apparatus of claim 18, wherein the merging order selector comprises:

a candidate node selector which selects 'P' nodes and 'S' nodes whose sub-nodes are labeled 'W' and 'B' as candidate nodes to be merged, wherein the 'P' nodes include the nodes at the predetermined depth having voxels located where objects exist and in the background, the 'S' nodes include the nodes having sub-nodes, the 'W' nodes include the nodes having all voxels located in the background, and the 'B' nodes include the nodes having all voxels located where objects exist;

an optimal node selector which selects, from among the candidate nodes as an optimal node, a node which can minimize a ratio of a difference ΔD between a number of distorted bits before merging the candidate nodes and a number of distorted bits after merging the candidate nodes with respect to a difference ΔR between a number of bits before merging the candidate bits and a number of bits after merging the candidate bits and labels the selected node 'B'; and

a candidate node updater which updates all the candidate nodes except the node selected as an optimal node.

20. (Previously Presented) The apparatus of claim 19, wherein the node encoder comprises:

a continue flag encoder which encodes a continue flag which includes information indicating whether or not a current node is an end of a compressed bitstream;

a node position encoder which encodes node position information indicating a position of each of the candidate nodes in a queue;

a node 'S'-or-'P' (SOP) selector which encodes node type information indicating whether or not the current node is an 'S' node or a 'P' node;

an S node encoder which encodes detailed information bit (DIB) data of an 'S' node if the node type information indicates that the current node is an 'S' node; and

a P node encoder which encodes DIB data of a 'P' node if the node type information indicates that the current node is a 'P' node.

21. (Previously Presented) The apparatus of claim 18, wherein the node encoder encodes only some of the merged nodes, ranging from a first merged node to a predetermined numbered merged node, are encoded.

22. (Original) The apparatus of claim 18, wherein the node encoder encodes all the merged nodes.

23. (Currently Amended) A method of decoding three-dimensional object data, comprising:

reading continue flag information from a bitstream of encoded three-dimensional object data and decoding the continue flag information;

decoding node type information of the bitstream, comprising:

decoding an 'S' node if the node type information indicates that a current node is a node having sub-nodes, and

decoding a 'P' node if the node type information indicates that the current node is a node at a predetermined depth of a tree structure having voxels located where objects exist and in a background and that the current node does not have sub-nodes; and

restoring the three-dimensional object data whose nodes are encoded to the tree structure.

24. (Currently Amended) A method of decoding three-dimensional object data, comprising:

decoding nodes of a bitstream of encoded three-dimensional object data, comprising decoding node type information of the bitstream, wherein the node type information describes nodes having sub-nodes and nodes at a predetermined depth of a tree structure having voxels located where objects exist and in a background, wherein the nodes at the predetermined depth of the tree structure having voxels located where objects exist and in the background do not have sub-nodes; and

restoring the three-dimensional object data whose nodes are encoded to the tree structure.

25. (Previously Presented) The method of claim 24, wherein decoding the nodes of the bitstream of the encoded three-dimensional object data further comprises:

reading continue flag information from a bitstream of encoded three-dimensional object data and decoding the continue flag information; and

reading node position information indicating which candidate node in a queue is a current node and decoding the node position information;

and wherein decoding node type information of the bitstream comprises:

decoding an 'S' node if the node type information indicates that a current node is a node having sub-nodes; and

decoding a 'P' node according to a prediction-by-partial-matching (PPM) algorithm if the node type information indicates that the current node is a node at the predetermined depth of the tree structure having voxels located where objects exist and in the background.

26. (Previously Presented) The method of claim 25, wherein in decoding the 'S' node, an average color of eight sub-nodes of the current node is decoded as detailed information bit (DIB) data, and the eight sub-nodes are sequentially decoded into nodes having all voxels located where objects exist ('B' nodes) or nodes having all voxels located in the background ('W' nodes).

27. (Previously Presented) The method of claim 26, wherein in decoding the 'P' node, the current node is PPM-decoded using DIB data bits, and red (R), green (G), and blue (B) values of 'B' voxels of the current node are decoded by

carrying out inverse adaptive arithmetic coding (AAC) and inverse differential pulse code modulation (DPCM).

28. (Currently Amended) An apparatus for decoding a bitstream of encoded three-dimensional object data, the apparatus comprising:

a bitstream reader which receives a bitstream of encoded three-dimensional object data;

a node decoder which decodes the bitstream, the node decoder comprising a node type selector which decodes node type information of the bitstream, wherein the node type information describes nodes having sub-nodes and nodes at a predetermined depth of a tree structure having voxels located where objects exist and in a background, wherein the nodes at the predetermined depth of the tree structure having voxels located where objects exist and in the background do not have sub-nodes; and

a tree structure restorer which restores decoded nodes to the tree structure.

29. (Previously Presented) The apparatus for claim 28, wherein the node decoder further comprises:

a continue flag decoder which decodes a continue flag indicating whether or not a current node is an end of the bitstream;

a node position information decoder which reads node position information indicating which candidate node in a queue is a current node and decodes the node position information;

and wherein the node type selector comprises:

an S node decoder which decodes an average color of eight sub-nodes of the current node as detailed information bit (DIB) data and then sequentially decodes the eight sub-nodes into nodes having all voxels located where objects exist ('B' nodes) or nodes having all voxels located in the background ('W' nodes), if the node type information indicates that the current node is a node having sub-nodes ('S' node); and

a P node decoder which prediction-by-partial-matching (PPM)-decodes DIB data of the current node and then decodes red (R), green (G), and blue (B) values of 'B' voxels of the current node by carrying out inverse adaptive arithmetic coding (AAC) and inverse differential pulse code modulation (DPCM) decoding, if the node type information indicates that the current node is a node at the predetermined depth of the tree structure having voxels located where objects exist and in a background ('P' node).

30. (Original) A computer-readable recording medium on which a program enabling the method of claim 1 is recorded.

31. (Original) A computer-readable recording medium on which a program enabling the method of claim 7 is recorded.

32. (Original) A computer-readable recording medium on which a program enabling the method of claim 23 is recorded.

33. (Original) A computer-readable recording medium on which a program enabling the method of claim 24 is recorded.